UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/830,164 | 04/21/2004 | Srikanth R. Avadhanam | 40062.0128USC1 | 8149 |

27488            7590            07/24/2009
MERCHANT & GOULD (MICROSOFT)
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903

| EXAMINER |
|---|
| PHAM, KHANH B |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2166 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 07/24/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

# BEFORE THE BOARD OF PATENT APPEALS
# AND INTERFERENCES

Application Number: 10/830,164
Filing Date: April 21, 2004
Appellant(s): AVADHANAM ET AL.

---

Gregory D. Leibold
For Appellant

## EXAMINER'S ANSWER

This is in response to the appeal brief filed 4/9/2009 appealing from the Office action

mailed 10/10/2008.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

No amendment after final has been filed.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

| | | |
|---|---|---|
| 6,438,562 | GUPTA | 8-2002 |
| 5,842,208 | BLANK | 11-1998 |

## (9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

### *Claim Rejections - 35 USC § 103*

1.      The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

2.      **Claims 1-26** are rejected under 35 U.S.C. 103(a) as being unpatentable over

Gupta et al. (US 6,438,562), and in view of Blank et al. (US 5,842,208), hereinafter

**"Gupta"** and **"Blank"**, respectively.

**As to claim 1, 12, 18, and 20,** Gupta teaches 'a method of creating an index for

a database table of records [col. 2, line 21-23, col. 3, line 45-47 fig 2-3], database table

corresponds to fig 2, table 200; index corresponds to fig 3, element 300 the method

occurring in a computer environment having a plurality of processing units [fig. 1, fig. 7]

wherein each processing unit has access to the table [col. 2, line 41-44], Gupta

specifically teaches relational storage where relational databases store data records in

indexed tables as detailed in fig 2-3, plurality of processing units corresponds to Gupta's

fig 1 and fig 7 ;

determining partition delimiters, each partition delimiter separating the table into

non-overlapping partitions of records [col. 14, line 35-38, fig 7], each partition delimiter

separating the table into non-overlapping partitions of records corresponds to Gupta's

fig 7, partitions A 161, B162, and C 163; 'each partition dedicated to one processing unit

for index creation' [col. 14, line 44-50, line 54-56], each partition dedicated to one

processing unit for index creation corresponds to Gupta's index fig 7, element 711, 712,

713, and 714 ;

wherein the step of determining comprises sampling the table records to

determine an approximate distribution of at least one key value in the record' [Col. 15,

line 35-47, line 66-67, Col. 16, line 1-13], Gupta specifically teaches sampling of "S"

records of the index maintenance records to compute good statistical representation of

the population chosen for "S" records, also suggested that every fifteenth record is

sampled during the PDML operations, it is also noted that "ranges are defined by

reading the "key values" associated with each multiple of S*/N from the sorted records

as detailed in fig 8, particularly "distributing work based on index key value ranges" [see

Col. 15];

accessing the table records in parallel,  [col. 8, line 1-13], Gupta teaches data

manipulation operations  specifically each data slave accessed to perform data

manipulation i.e., processing data records and updating the index maintenance records

as detailed in col. 8, line 1-13;

filtering the accessed records in parallel, wherein each processing unit

determines which records to keep '[col. 7, line 45-51; col. 12, line 21-27, col. 13, line 37-

39], Gupta specifically teaches accessing index ID value that identifies the specific

index associated with the data records to be changed as detailed in col. 12, line 21-27

independently creating a plurality of sub-indexes, wherein at least two sub-indexes are

created by different processing units' [col. 3, line 45-52, col. 12, line 58-63,.line 64-67,

col. 13, line 18-25, col. 14, line 54-61], Gupta specifically teaches each index

record corresponds to a row [see fig 6], further index maintenance records indicate

changes that need to be made to indexes in response to changes that are made to the

table [col. 12, line 58-63],that corresponds to independently creating indexes or sub-

indexes, further to keep separate the changes to the two indexes, the index maintained

records are modified to include the index ID as detailed in fig 6, element 611. It is also

noted that sub-indexes are part of B-tree element 300 because B-tree is arranged in

hierarchical structure, further each node or branch in the B-tree structure associated

with index key [col. 3, line 45-53];

'storing the final index' [col. 20, line 57-60], Gupta specifically teaches storing

index records related to global index particularly sorted version of the index

maintenance records as detailed in col. 20, line 57-60.


It is however, noted that Gupta does not specifically teach 'merging the

sub-indexes together to create, a final index related to the table' nor 'each processing

unit accesses all of the records in the table of records', although Gupta

specifically teaches coordinating an update of a global index of an indexed table and

updates the global index using the current index maintenance record [col. 7, line 41-42],

further Gupta also suggests index maintenance records using "data manipulation

operations among parallel data manipulation slaves for example fig 5, element 510, the

data manipulation operations including updating col. 14, line 46-47], inserting, deleting [col. 13, line 28] sorting [col. 16, line 36-37]and like

On the other hand, Blank et al. disclosed 'each processing unit accesses all of the records in the table of records' at Col. 1 lines 17-24 and 'merging the sub-indexes together to create, a final index related to the table' col. 3, line 57-67, col. 4, line 1-2], Blank specifically teaches index built system that supports multiple scan program performed in parallel by multiple processors against multiple partitions [see fig 3], further, in the processing or recover/built index system, multiple merge programs for example fig 4, element 112a-b are performed that merges all the key/rid values. Finally, an index built program 114 is performed to built final index element 116as detailed in fig 1. It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank et al. into parallel index maintenance of Gupta et al. because, both Gupta and Blank are directed to "multiple processors and multiple partitions of database tables [see Gupta fig 1, fig 7; Blank: fig 1, element 102 corresponds to multiple processors, element 120 corresponds to partition], both Gupta and Blank also teaches "indexing and index key [see Gupta: fig 2, col. 13, line 26-40; Blank: col. 2, line 42-48] and both Gupta and Blank specifically suggests "sort" operation [see Gupta: col. 16, line 31-33; Blank: col. 3, line 18-19] and both are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank into parallel index maintenance of Gupta et al. because that would have allowed user's of Gupta to use "sort program that executing in parallel receive the san streams for each "partition to create sort stream [col. 3, line 5-9], while merge program that merges the sort stream received from the sort program to create a merge stream [col. 3, line 10-15], further merge program built "final indexes" col. 3, line 67, col. 4, line 1-2] bring the advantages of "high performance recover/build index system that reduces the amount of time that takes to built index in multiple processors [col. 3, line 25-27], furthermore, piping the data between the sort and merge programs improves performance of the system as suggested by Blank et al. [col. 3, line 28-30].

**As to claim 2**, Gupta disclosed 'wherein the act of creating the sub-indexes [Col. 3, line 45-53], sub-indexes are part of B-tree element 300 because B-tree is arranged in hierarchical structure, further each node or branch in the B-tree structure associated with index key [Col. 3, line 45-53] further comprises sorting the records and generating a data structure based on the sorted records [Col. 8, line 18-26].

**As to claim 3**, Gupta disclosed wherein the data structure is a B-Tree data structure [Col. 3,line 45-448, Col. 4, line 13-14], B-structure data structure corresponds to Gupta's B-tree fig 3, element 300.

As to claim 4, Gupta disclosed 'wherein the data structure has multiple levels. [fig 3, element 300, Col. 4, line 13-15], B-tree data structure is a hierarchical having root node, leaf nodes.

As to claim 5, Gupta disclosed 'wherein the data structure is a clustered index' [Col. 14, line 23-26], Gupta specifically teaches index will be clustered based on index maintenance records.

As to claim 6, Gupta disclosed 'further comprising gathering sub-index statistical information and stitching sub-index statistical information' [Col. 15, line 35-50, fig 5], Gupta specifically suggests sample of "S" records of the index to give good statistical representation of the population based on number of available nodes as detailed in [Col. 15, line 35-47].

As to claim 7, Gupta disclosed 'wherein the method is initiated by an index creation manager module' [fig 1, element 170,fig 7, element 170], global index corresponds to index module.

As to claim 8, Gupta disclosed 'wherein the method is initiated by a query manager in response to a supplied query' [fig 13, Col. 20, line 66-67, Col. 21, line 1].

**As to claim 9**, Gupta disclosed 'wherein the method is initiated automatically in response to a modification to the table' [Col. 5, line 38-44, Col. 18, line 53-57, fig 11].

**As to claim 10**, Gupta disclosed 'wherein the act of determining partition delimiters comprises: creating a histogram based on the sampled information; and evaluating the histogram to determine the partition delimiters [Col. 15, line 39-40].

As best understood by the examiner, a histogram can be constructed by segmenting the range of the data into equal sized, particularly, ranges that are defined in Col. 15, line 66-67, moreover, it is common knowledge that statistics analyzing, viewing the data in a variety of ways, one possible way is "histogram", "bar graphs", "pie-charts", further, "histograms are sometimes referred to "frequency distribution" which is an integral part of Gupta's "statistical representation of records [Col. 15, line 39-40]

**As to claim 11, 13**, Gupta disclosed 'determining a processor goal value based on the number of processors in the computer system' [Col. 4, line 52-55]; determining a least common multiple value based on the processor goal value [Col. 6, line 55-59]; 'determining whether the histogram information may be substantially evenly split into the least common multiple value number of partitions' [Col. 6, line 59-65,Col. 13, line 57-61]; if so, creating the partition delimiters based on the least common multiple value' [Col. 13, line 66-67]; and if not, adjusting the processor goal to determine a new least common multiple value to determine partition delimiters' [Col. 14, line 3-8].

**As to claim 14**, Gupta teaches a system which including 'a system for database

table index creation for a database table [fig 1, Col. 4, line 57-61], database table

corresponds to fig 1, database table], the database table stored in memory and

comprising a plurality of records [fig 1-2, element 151-153], the system comprising:

a partition tool that determining partition delimiters, each partition delimiter

separating the table into non-overlapping partitions of records [col. 14, line 35-38, fig 7],

each partition delimiter separating the table into non-overlapping partitions of records

corresponds to Gupta's fig 7, partitions A 161, B162, and C 163; 'each partition

dedicated to one processing unit for index creation' [col. 14, line 44-50, line 54-56], each

partition dedicated to one processing unit for index creation corresponds to Gupta's

index fig 7, element 711, 712, 713, and 714 ;

wherein the step of determining comprises sampling the table records to

determine an approximate distribution of at least one key value in the record' [Col. 15,

line 35-47, line 66-67, Col. 16, line 1-13], Gupta specifically teaches sampling of "S"

records of the index maintenance records to compute good statistical representation of

the population chosen for "S" records, also suggested that every fifteenth record is

sampled during the PDML operations, it is also noted that "ranges are defined by

reading the "key values"  associated with each multiple of S*/N from the sorted records

as detailed in fig 8, particularly "distributing work based on index key value ranges" [see

Col. 15];

a plurality of processing units that respectively accesses the database table in

parallel, [fig 1, Col. 4, line 43-48] wherein each of the respective processing units

accesses each of the records [Col. 8, line 1-13], Gupta teaches data manipulation

operations specifically each data slave accessed to perform data manipulation i.e.,

processing data records and updating the index maintenance records as detailed in Col.

8, line 1-13;


and 'filters the accessed records to determine which records to keep'[Col. 7, line

45-51; Col. 12, line 21-27, Col. 13, line 37-39], Gupta specifically teaches accessing

index ID value that identifies the specific index associated with the data records to be

changed as detailed in Col. 12, line 21-27;


'wherein each of the respective processing units creates a sub-index of

database table records resulting in a plurality of sub-indexes'; [Col. 3, line 45-52, Col.

12, line 58-63, line 64-67, Col. 13, line 18-25, Col. 14, line 54-61], Gupta specifically

teaches each index record corresponds to a row [see fig 6], further index maintenance

records indicate changes that need to be made to indexes in response to changes that

are made to the table [Col. 12, line 58-63],that corresponds to independently creating

indexes or sub-indexes, further to keep separate the changes to the two indexes, the

index maintained records are modified to include the index ID as detailed in fig 6,

element 611. It is also noted that sub-indexes are part of B-tree element 300 because

B-tree is arranged in hierarchical structure, further each node or branch in the B-tree

structure associated with index key [Col. 3, line 45-53]

'a store tool that stores the final database table index' [Col. 20, line 53-62].


It is however, noted that Gupta et al. does not specifically teach 'merge tool that

merges the plurality of sub-indices into a final database table index', although Gupta

specifically teaches coordinating an update of a global index of an indexed table and

updates the global index using the current index maintenance record [Col. 7, line 4i-42],

further Gupta also suggests index maintenance records using "data manipulation

operations among parallel data manipulation slaves for example fig 5, element 510, the

data manipulation operations including updating Col. 14, line 46-47], inserting, deleting

[Col. 13, line 28] sorting [Col. 16, line 36-37]


On the other hand, Blank et al. disclosed 'merge tool that merges the plurality of

sub-indices into a final database table index' Col. 3, line 57-67, Col. 4, line 1-2], Blank

specifically teaches index built system that supports multiple scan program performed in

parallel by multiple processors against multiple partitions [see fig 3], further, in the

processing or recover/built index system, multiple merge programs for example fig 4,

element 112a-b are performed that merges all the key/rid values. Finally, an index built

program 114 is performed to built final index element 116 as detailed in fig 1.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank et al. into parallel index maintenance of Gupta et al. because, both Gupta and Blank are directed to "multiple processors and multiple partitions of database tables [see Gupta fig 1, fig 7; Blank: fig 1, element 102 corresponds to multiple processors, element 120 corresponds to partition], both Gupta and Blank also teaches "indexing and index key [see Gupta: fig 2, Col. 13, line 26-40; Blank: Col. 2, line 42-48] and both Gupta and Blank specifically suggests "sort" operation [see Gupta: Col. 16, line 31-33; Blank: Col. 3, line 18-19] and both are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank et al. into parallel index maintenance of Gupta et al. because that would have allowed user's of Gupta et al. to use "sort program that executing in parallel receive the san streams for each "partition to create sort stream [Col. 3, line 5-9], while merge program that merges the sort stream received from the sort program to create a merge stream [Col. 3, line 10-15], further merge program built final indexes" Col. 3, line 67, Col. 4, line 1-2] bring the advantages of "high performance recover/build index system that reduces the amount of time that takes to built index in multiple processors [Col. 3, line 25-27], furthermore, piping the data between the sort and merge programs improves performance of the system as suggested by Blank et al. [Col. 3, line 28-30],

As to claim 15, Gupta disclosed 'a filter module that filters the accessed records and selectively predetermined records"[Col. 7, line 45-51; Col. 12, line 21-27, Col. 13, line 37-39], Gupta specifically teaches accessing index ID value that identifies the specific index associated with the data records to be changed as detailed in Col. 12, line 21-27, Col. 20, line 66-67, Col. 21, line 1-4, fig 13] ; and a sorting module that sorts records kept by the filter module into a sub-index' [Col. 16, line 31-33]. On the other hand, Blank disclosed 'a scanning module that scans the database table' [fig 1, element 108,fig 2, element 200], Blank specifically teaches both scan and sort operations as detailed in fig 2.

As to claim 16, Blank disclosed 'scanning module, filter module and sorting module, for each processing unit, operate concurrently' [fig 1-2,fig 4,Col. 3, line 55-67].


As to claim 17, Gupta disclosed 'a sampling module for sampling the database table and a partition module for dividing the records into substantially equal quantities related to the number of processing units' [Col. 15, line 35-47].


As to claim 19, Gupta disclosed 'upon determining that the accessed table record is not associated with the at least one partition dedicated to the first processing unit, passing the accessed record to the second processing unit for index creation' [Col. 16, line 34-46].

**As to claim 21, 25,** Gupta disclosed wherein the act of allocating portions of the

disk allocates a predetermined number of blocks, the predetermined number of blocks

is determined during the determination of the partition delimiters' [Col. 11, line 61-67,

Col. 12,line 1-7].

**As to claim 22, 26,** Gupta disclosed 'wherein the allocation of portions of the

disk comprises: maintaining a cache of allocated pages and allocating pages for each

partition in the cache for each processing unit' [Col. 3, line 6-15, fig 1]

retrieving a pre-determined number of database pages upon request, Col. 3,

line 15-18] wherein the number of pages to allocate upon each request is determined by

the size of the cache [Col. 3, line 19-26].

**As to claim 23,** Gupta disclosed 'wherein the cache has a size depending

on the size of the index being built and the number of currently available free pages in

the system' [Col. 6, line 24-33].

**As to claim 24,** Gupta teaches a system which including 'In a computer system

having a plurality of processors' [fig 1, element 111,112,113,114], an index creation

system for creating an index of information for a table of data records' [fig 1, element

170] 'a sampling module that samples the table of data records to determine sub

index delimiters' [[Col. 15, line 35-47, line 66-67, Col. 16, line 1-13], Gupta specifically

teaches sampling of "S" records of the index maintenance records to compute good

statistical representation of the population chosen for "S" records, also suggested that

every fifteenth record is sampled during the PDML operations, it is also noted that

"ranges are defined by reading the "key values" associated with each multiple of S*/N

from the sorted records as detailed in fig 8, particularly "distributing work based on index

key value ranges" [see Col. 15]; further it is noted that Gupta also specifically teaches

"partitioned" database tables as detailed in fig 1 and fig 7;

wherein the sub-index delimiters are used as partition delimiters separating the

table into non-overlapping portion of record' [col. 14, line 35-38, fig 7], each partition

delimiter separating the table into non-overlapping partitions of records corresponds to

Gupta's fig 7, partitions A 161, B162, and C 163


' two or more index creation modules, each index creation module associated

with a processor, each index creation module creates a sub-index'

[Col. 3, line 37-65, Col. 4, line 13-24]i


an access module that accesses each of the data records from the table of data

records [Col. 8, line 1-13], Gupta teaches data manipulation operations specifically each

data slave accessed to perform data manipulation i.e., processing data records and

updating the index maintenance records as detailed in Col. 8, line 1-13;


'a filter module that filters data records according the sub-index

delimiters to keep only relevant data records' '[Col. 7, line 45-51; Col. 12, line 21-27,

Col. 13, line 37-39], Gupta specifically teaches accessing index ID value that identifies

the specific index associated with the data records to be changed as detailed in Col. 12,

line 21-27

'a sorting module that sorts the relevant data records into a sub- index' [Col. 3,

line 45-53], sub-indexes are part of B-tree element 300 because B-tree is arranged in

hierarchical structure, further each node or branch in the B-tree structure associated

with index key [Col. 3, line 45-53] further comprises sorting the records and generating

a data structure based on the sorted records [Col. 8, line 18-26].

'a store module that stores the final index' [Col. 20, line 56-60].

It is however noted that Gupta does not specifically teach 'a merge module that

merges the sub-indexes into a final index', although Gupta specifically teaches

coordinating an update of a global index of an indexed table and updates the global

index using the current index maintenance record [Col. 7, line 41-42], further Gupta also

suggests index maintenance records using "data manipulation operations among

parallel data manipulation slaves for example fig 5, element 510, the data manipulation

operations including updating Col. 14, line 46-47], inserting, deleting [Col. 13, line 28]

sorting [Col. 16, line 36-37].

On the other hand, Blank et al. disclosed 'a merge module that merges the sub-indexes into a final index" Col. 3, line 57-67, Col. 4, line 1-2], Blank specifically teaches index built system that supports multiple scan program performed in parallel by multiple processors against multiple partitions [see fig 3], further, in the processing or recover/built index system, multiple merge programs for example fig 4, element 112a-b are performed that merges all the key/rid values. Finally, an index built program 114 is performed to built final index element 116as detailed in fig 1.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank et al. into parallel index maintenance of Gupta et al. because, both Gupta and Blank are directed to "multiple processors and multiple partitions of database tables [see Gupta fig 1, fig 7; Blank: fig 1, element 102 corresponds to multiple processors, element 120 corresponds to partition], both Gupta and Blank also teaches "indexing and index key [see Gupta: fig 2, Col. 13, line 26-40; Blank: Col. 2, line 42-48] and both Gupta and Blank specifically suggests "sort" operation [see Gupta: Col. 16, line 31-33; Blank: Col. 3, line 18-19] and both are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Blank into parallel index maintenance of Gupta et al. because that would have allowed user's of Gupta to use "sort program that executing in parallel receive the san streams for each "partition to create sort stream

[Col. 3, line 5-9], while merge program that merges the sort stream received from the

sort program to create a merge stream [Col. 3, line 10-15], further merge program built

"final indexes" Col. 3, line 67, Col. 4, line 1-2] bring the advantages of "high

performance recover/build index system that reduces the amount of time that takes to

built index in multiple processors [Col. 3, line 25-27], furthermore, piping the data

between the sort and merge programs improves performance of the system as

suggested by Blank [Col. 3, line 28-30].

## (10) Response to Argument

## A. Response to appellant's argument that "The cited Gupta and Blank references do not, alone or in combination, teaches all of the claimed limitations of the independent claims"

<u>Regarding claim 1</u>

The examiner agrees with Appellant's statement at page 8 of the Appeal Brief

that "Gupta teaches distributing the records among multiple slave processes based

upon the ranges they fall into. The slave processes then perform maintenance only on

the records which they receive". Gupta therefore teaches **accessing the table records**

**in parallel**, utilizing multiple slave processes.  The examiner also agreed with Appellant

that Gupta does not teach "each processing unit access all of the records in the table of

records."

Regarding the Blank reference, the examiner agrees with Appellant's statement at page 8 that Blanks relates to a "recover/build index system that builds an index for a file by scanning **partitions of the file in parallel** to retrieve key values and their associated record identifier (rid) values" (Blank, Abstract) . As seen in Blank's Fig. 1 reproduced below, the **partitions of the file** 118 comprises all of the records , Blank therefore teaches "accessing the table records in parallel, wherein each processing unit accesses **all of the records** in the table of records" as claimed.
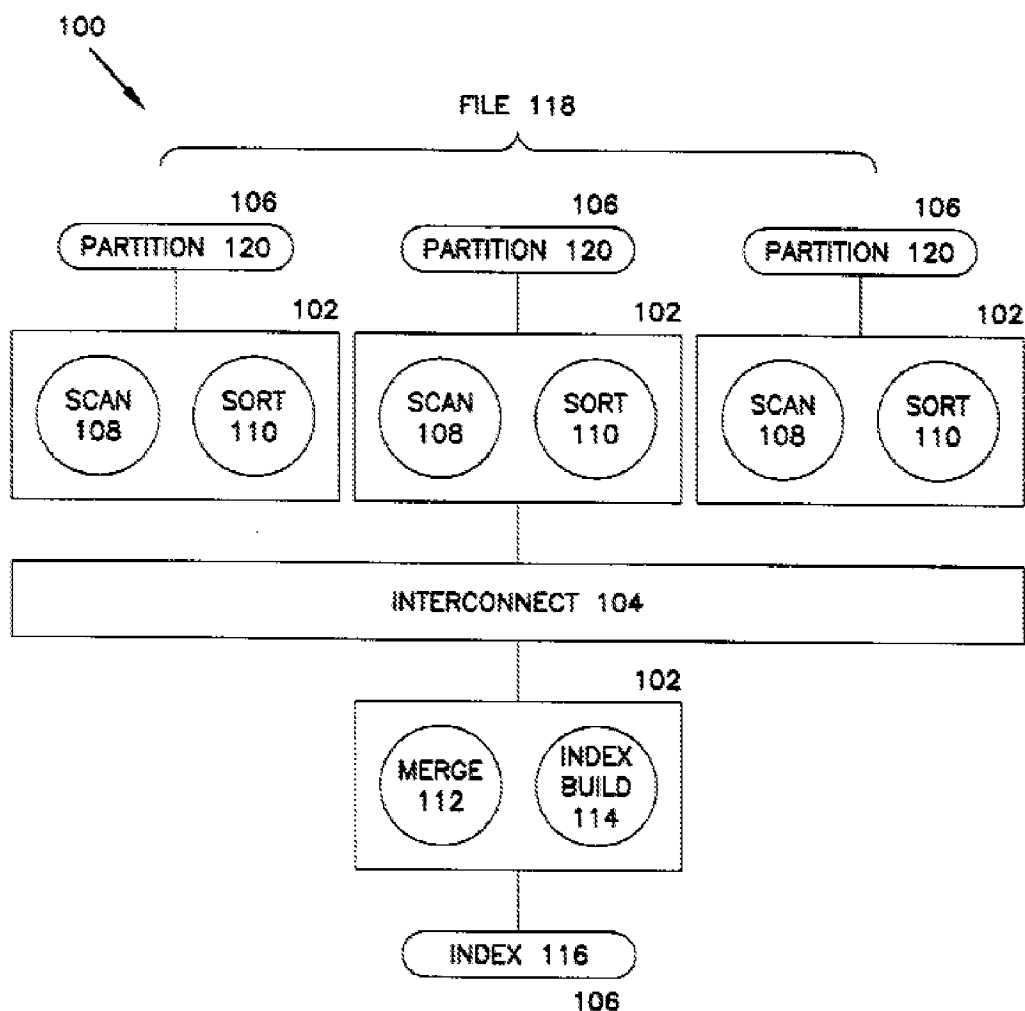
Appellant further argued that Blank does not teach "accessing the table records in parallel" because Blank teaches "the index building steps are performed **serially**". On the contrary, as seen in Fig. 1 above, Blank clearly teaches at Col. 2 lines 17-25 that "The present invention is typically implemented using a number of computer programs **executed in parallel by the processors 102**", wherein each processor 102 corresponds to the claimed "processing unit".

Appellant further argued at page 9 that "while **in Blank all records are scanned**, it does not necessarily follow that each of the scan programs in Blank scan all of the records." The examiner respectfully submits that the claim only requires that "each processing units **accesses**[1] all of the records", but does not require each processing units **scan** all of the records. As seen in Fig. 1, the file 118 comprises all records from a plurality of partitions 120. Therefore, accessing the file 118 is same as "accessing all of the records," and in order to access and/or scan each partition, each of the processor must access the file comprising all records.

Further, claim 1 of Blank reference provides the step of "performing, in the computer, **multiple scans in parallel against the file...",** which indicates that each of the scan program access the file comprises all records as claimed.


Regarding claim 14.

---

[1] Microsoft Press Computer Dictionary, 3rd edition defines ACCESS as: "To gain entry to memory in order to read or write data".

Appellant's argument regarding claim 14 is similar to claim 1. The examiner's

respond to claim 1 is also applied to this claim.


## Regarding claim 18

Appellant's argument regarding claim 18 is similar to claim 1. The examiner's

respond to claim 1 is also applied to this claim.

Appellant further agued that Blank does not teach "storing a result produced by

the first processing unit for later use in locating records". On the contrary, Blank teaches

this limitation at Col. 3 lines 35-39, which states: "after scanning the partitions 120 to

extract key values and record identifiers, this information is **written to a file** on a data

storage device 106. Then in case of system failure, the index 116 can be quickly rebuilt

using the steps discussed above".


## Regarding claim 20

Appellant's argument regarding claim 20 is similar to claim 1. The examiner's

respond to claim 1 is also applied to this claim.


## Regarding claim 24.

Appellant's argument regarding claim 24 is similar to claim 1. The examiner's

respond to claim 1 is also applied to this claim.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the

Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Khanh B. Pham/

Primary Examiner

Art Unit 2166.


Conferees:

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166


/Eddie C. Lee/

Supervisory Patent Examiner, TC 2100